

Netzwerkanalyse zum Disney- und Pixar-Universum

J. Marquez, M. Sager, P. Bort, C. Kuba, F. Arnold

R-Code mit Erläuterungen

Disney und Pixar sind mitunter die größten Filmstudios für Zeichentrick- und Animationsfilme. Viele Filmfiguren werden wiederverwertet usw. Daher galt unser Forschungsinteresse der Frage, in welchen Filmen treten gehäuft Cameos auf, welche Figuren kommen immer wieder vor und lässt sich eine Korrelation nach Produktionsbudgets ableiten?

Die Entwicklung des Netzwerks

Im ersten Schritt soll gezeigt werden, wie die Daten zu den wichtigsten Filmfiguren und Mitarbeitern erhoben und für die Verwendung in RStudio aufbereitet wurden.

Bei diesem Projekt entschieden wir uns zu jedem Film die wichtigsten Figuren mit zu erfassen, selbst wenn diese keine Cameo-Auftritte in anderen Filmen haben. Das hatte den Vorteil, dass eine Clusterung der Figuren nach Filmen möglich ist und sich die einzelnen Filme enger zusammenrücken. Im Gegenzug bedeutete das aber auch einen erheblichen Mehraufwand bei der Erfassung der Daten. Schlussendlich wuchs die Edgelist auf über 3000 Verbindungen.

Quellen für das Projekt:

https://pixar.fandom.com/wiki/Pixar_Wiki
https://pixar.fandom.com/wiki/References_to_Upcoming_Films
https://disney.fandom.com/wiki/The_Disney_Wiki
<https://www.imdb.com/>
<https://en.wikipedia.com>
<https://www.gamesradar.com/50-best-pixar-easter-eggs/>
<https://www.insider.com/pixar-movie-easter-eggs-2017-1>
https://en.wikipedia.org/wiki/List_of_Pixar_film_references

Optionen für die Arbeitsumgebung

Leider bringt ein großes Netzwerk die R-Studio Umgebung sehr schnell an Grenzen. Standardmäßig limitiert R-Studio die Ausgabe von Variablen auf etwa 400. Durch die enorme Größe unseres Netzwerks können bei den Default-Einstellungen nicht alle Verbindungen dargestellt werden. Dafür muss mit dem folgenden Befehl die maximale Anzahl an Darstellungen erhöht werden.

```
options(max.print=999999)
```

Einlesen und Überprüfen der Daten

Einlesen der Filme und Figuren Die Edge und Nodelist der Figuren und Filme wird über read.csv von Github geladen

```
edges <- read.csv("https://raw.githubusercontent.com/tinomarst/pixar-226305/master/daten/el_fig.csv",  
                  header=T, as.is=T, sep=",") #Die Edgelist wird von Github geladen  
nodes <- read.csv("https://raw.githubusercontent.com/tinomarst/pixar-226305/master/daten/nl_fig.csv",  
                  header=T, as.is=T, sep = ",") #die Nodelist wird von Github geladen
```

```

matrix <- as.matrix(edges) #Matrix erstellen
#Gesamtnetzwerk der Figuren wird aus der Edge- Und Nodelist gelesen#
#Die Daten werden im Dataframe gespeichert
disnet <- graph_from_data_frame(d=matrix, vertices=nodes, directed=FALSE)

#Gesamtnetzwerk wird für einen Rohplot nochmals eingelesen
#was für die Darstellung eines unserer Netzwerke wichtig war.
disraw <- graph_from_data_frame(d=matrix, vertices=nodes, directed=FALSE)

```

Einlesen der Mitarbeiter Als zweite Datenquelle wird auch die Edge und Nodelist der Mitarbeiter und Filme mit dem Befehl `read.csv` von Github geladen. Aufgrund der Fülle an Daten und der damit verbundenen Berechnungsdauer verwenden wir diese Listen nicht weiter zur Darstellung. Trotzdem lässt sich damit zeigen, dass es auch möglich ist ein zweites Netzwerk in R-Studio darzustellen.

Werte Überprüfen und Netzwerk validieren

Da es zu Beginn unserer Arbeiten Unstimmigkeiten zwischen der Darstellung des Netzwerks und den Daten in der Edge- und Nodelist gab, war der erste Schritt eine Überprüfung der Daten, die R-Studio in der Matrix speicherte. Für einen reinen Plot der Netzwerke kann dieser Teil auch auskommentiert werden. Wir blenden die Ausgabe aus, indem wir den Befehl `results='hide'` verwenden. **Edge- und Vertex-Attribute der Figuren anzeigen**

```

list.vertex.attributes(disnet)
list.edge.attributes(disnet)

```

Auch bei den Befehlen, die zu Beginn eines Chunks stehen, können mehrere Einstellungen gleichzeitig vorgenommen werden. Dabei werden die verschiedenen Begriffe mit einem Komma getrennt.

```

***Edge- und Vertex-Attribute der Mitarbeiter anzeigen**
list.vertex.attributes(disnetma)
list.edge.attributes(disnetma)

```

Überprüfung der Node-Daten der Figuren

```

V(disnet)$name #Name des jeweiligen Knotens aus der Figuren-Nodelist
V(disnet)$art # Art des jeweiligen Knotens aus der Figuren-Nodelist
V(disnet)$prod # Art des jeweiligen Knotens aus der Figuren-Nodelist
V(disnet)$year # Art des jeweiligen Knotens aus der Figuren-Nodelist
V(disnet)$income # Art des jeweiligen Knotens aus der Figuren-Nodelist

```

Durch einen händischen Vergleich zeigte sich bald, dass große Teile der Daten fehlten. Durch einige Anpassungen der Edge- und Nodelist sowie die Erweiterung des Darstellungsspeichers von R-Studio konnten nun sämtliche Daten fehlerfrei in eine Matrix überführt werden.

Plot des Rohnetzwerks

Ausgabe des Gesamtnetzwerks der Figuren ohne grafische Anpassung In diesem Schritt plotten wir das Gesamtnetzwerk um einen Eindruck von der Größe des Netzwerks zu gewinnen. Hier

Bei den Betweenness Werten stellte sich vor allem der Film “Ralph Breaks The Internet” als besonders interessant heraus.

Ausgabe der Statistiken zu den Mitarbeitern

Mit den selben Befehlen, wie bei den Figures erheben wir auch die statistischen Daten zu den Mitarbeitern der Studios. Durch die Auswertung der Netzwerkmaße kommen wir zu dem Schluss, dass wir folgende Mitarbeiter in Ego und Teilnetzwerken weiter untersuchen wollen: Clyde Geronimi, Hamilton Luske, Wilfred Jackson, Walt Disney, Wolfgang Reitherman, Bill Peet, Erdman Penner, John Lasseter, Ralph Wright, Ted Sears, Vance Gerry

Einstellungen zur grafischen Darstellung des Netzwerks

Da uns trotzdem die allgemeine Darstellung des Gesamtnetzwerks interessiert, müssen wir unsere Visualisierung mit ein paar Befehlen anpassen. Wie sich schon in der Rohversion des Gesamtnetzwerks gezeigt hat, kann mit einem riesen Hairball nicht gearbeitet werden. Daher ist eine Anpassung der Darstellung notwendig, mit der die einzelnen Knoten unterschieden werden können. Im gleichen Schritt wird eine Entzerrung des Netzwerks vorgenommen.

Die Anpassung haben wir in diesem Schritt explizit für den Knit-Vorgang in eine PDF-Datei vorgenommen.

Grafische Anpassung der Knoten und Kanten des Gesamtnetzwerks

Da wir beim Einlesen der Daten zwei Netzwerke erstellen, müssen wir auch beiden Netzwerken Eigenschaften für die Knoten und die Kanten zuweisen. Das ist zwar doppelte Arbeit, aber dadurch können wir beide Netzwerke getrennt voneinander bearbeiten.

Es gibt für solche Fälle in iGraph auch eine Funktion, mit der die Einstellungen schon zu Beginn des Dokuments global für alle Netzwerke festgelegt werden kann.

```
E(disnet)$width <- 2.5 #Die Kanten des Netzwerks sollen generell eine Stärke  
#von 2.5 haben. Damit ist eine gute Sichtbarkeit bei kleinen Netzwerken gewährleistet.  
#Soll der Wert abweichen kann das inline, also direkt im Plot-Befehl für den jeweiligen  
#Plot angepasst werden. Schlussendlich haben wir alle Daten einzeln anpassen müssen.
```

```
#Das lag daran, dass die Darstellung in RStudio von der Darstellung in der durch KNIT  
#erstellten PDF abweicht.
```

```
V(disnet)$size <- 4 #Hier legen wir auch die Größe der Knoten für alle Plots fest,  
#die aus dem Netzwerk 'disnet' generiert werden. In vielen Fällen mussten wir die  
#Größe für den Plot in die PDF-Datei nochmals gesondert einstellen.
```

```
# Einstellungen für das Disnet-Netzwerk  
#Hier legen wir für das Disnet Netzwerk fest, dass sämtliche Filme als  
#Rechteck "square" und Figuren als Kreise "circle" angelegt werden.
```

```
V(disnet)[V(disnet)$art == 1]$shape <- "square" #Gesamtnetzwerk Figuren  
V(disnet)[V(disnet)$art == 2]$shape <- "circle"  
V(disnet)[V(disnet)$art == 3]$shape <- "circle"  
V(disnet)[V(disnet)$art == 4]$shape <- "circle"  
V(disnet)[V(disnet)$art == 5]$shape <- "circle"  
V(disnet)[V(disnet)$art == 6]$shape <- "circle"  
V(disnet)[V(disnet)$art == 7]$shape <- "circle"  
V(disnet)[V(disnet)$art == 8]$shape <- "circle"
```

```

V(disnet)[V(disnet)$art == 9]$shape <- "circle"
V(disnet)[V(disnet)$art == 99]$shape <- "circle"
# Die Zuweisung sieht etwas kompliziert aus, ist aber gar nicht so schwer. Außerdem
#braucht man diesen Befehl regelmäßig!
# Zur Erklärung: Der Befehl sagt: Wenn der Wert $art aus dem
#Disnet-Netzwerk [V(disnet)$art == 99] den entsprechenden Wert hat, dann weise den
#Knoten V(disnet) die Form "circle" als $shape zu.
# Auf Deutsch - wenn $art den Wert 1 hat, sollen die Vertices als Rechteck
#dargestellt werden.

#Auch die Farben lassen sich bequem im Voraus festlegen.
#Das Prinzip ist das gleiche. Hier beziehen wir uns allerdings auf den Wert $prod
#als Produktionsstudio aus der Nodelist. Über den $color Befehl lässt sich dann eine
#spezifische Farbe zuweisen.
#Eine gute Liste an Farben für R-Studio gibt es auf
# http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf
V(disnet)[V(disnet)$prod == 1]$color <- "cornflowerblue"
V(disnet)[V(disnet)$prod == 2]$color <- "cornflowerblue"
V(disnet)[V(disnet)$prod == 3]$color <- "green"
V(disnet)[V(disnet)$prod == 4]$color <- "red"
V(disnet)[V(disnet)$prod == 5]$color <- "black"
V(disnet)[V(disnet)$prod == 8]$color <- "cornflowerblue"
V(disnet)[V(disnet)$prod == 6]$color <- "yellow"
V(disnet)[V(disnet)$prod == 99]$color <- "white"

V(disnetma)[V(disnetma)$art == 1]$shape <- "square" # Die Einstellungen müssen für das
#Gesamtnetzwerk Mitarbeiter separat vorgenommen werden. Daher können die oberen
#Einstellungen prinzipiell kopiert werden. Allerdings muss dann der Name des Netzwerks
#angepasst werden.

V(disnet)$name <- nodes$name
#Dieser Schritt ist extrem wichtig. Hiermit werden die jeweiligen Bezeichnungen aus der
#Namen-Spalte der Nodelist übernommen. Wird der Name nicht ersetzt, zeigt der Plot immer
#die ID aus der Edgelist an.
#Theoretisch lassen sich auch andere Werte (nicht nur aus der Nodelist) anzeigen. Auch
#Akteursmaße, wie den Betweenness- oder Degree-Wert könnt ihr so anzeigen lassen. Aber
#in der Regel wollt ihr den Namen sehen.

```

Disney-Pixar Gesamtnetzwerk der Figuren

In diesem Schritt sind die ersten optischen Einstellungen soweit vorgenommen, dass sich die Knoten im Gesamtnetzwerk unterscheiden lassen.

```
#Hier erfolgt auch die Erklärung der meisten Einstellungen um den Plot-Befehl.

#Zunächst wird eine neue Variable mit dem Namen Durchmesser erzeugt.
#Der weisen wir das zu verwendende Layouts für den Plot zu.
#Durch diesen Befehl kann man beim Plotten gezielt in das Netzwerk hineinzoomen.
#Allerdings werden bei einem Wert von mehr als 1.0 die Ränder des Netzwerks abgeschnitten.
durchmesser <- layout_with_kk(disnet)
durchmesser <- norm_coords(durchmesser, ymin=-1, ymax=1, xmin=-1, xmax=1)

#Deshalb weisen wir dem Plot eine weitere Eigenschaft zu, die einen Plot bis an die
#äußeren Ränder der späteren PDF ermöglicht. Diese Einstellung haben wir erst wesentlich
#später gefunden. Für den Plot in eine PDF ist sie unerlässlich und macht die
#"durchmesser"-Variable quasi überflüssig.

# "par" legt den Bereich für einen Plot fest. Die Einstellung "mfrow" bezeichnet eine
#Matrix, in der die Plots später angezeigt werden. 1,1 bedeutet, 1 einzelner Plot pro
#Reihe und Spalte. 1,2 bedeutet zwei Plots nebeneinander usw.
#Die Einstellung "mar" gibt an, wo die Ränder des Plots liegen sollen.
#Standard ist (1,1,1,1). Als schöner hat sich für uns der Wert (0,0,0,0) erwiesen.
#Dabei werden die Ränder nach folgendem Schema festgelegt: (unten,links,oben,rechts).
# Ein Wert < 0 ist nicht möglich
par(mfrow=c(1,1), mar=c(0,0,0,0))

#Nun folgt der Plot Befehl mit einigen wichtigen Einstellungen
plot(disnet, #Plotte das Netzwerk "disnet"
      vertex.size=0.3, #ändere die Größe der Knoten auf den Wert 0.3 (der war passend
#für unser Gesamtnetzwerk) normalerweise ist ein Wert zwischen 3 und 20 optimal.
      vertex.color="orange", #Mit diesem Befehl wird die Farbe der Knoten angepasst.
      vertex.label.cex=0.1, # Ändert die Schriftgröße der Beschriftung der Knoten
      vertex.frame.width=0.001, #Verändert die Stärke des Rahmen um einen Knoten. Normal 1
      rescale=F, #Wichtig, wenn das Layout zuvor mit einer Variable (hier durchmesser)
#vergeben wurde. Sollen mehrere Netzwerke unabhängig voneinander gezoomt werden,
#sind jeweils eigene Variablen notwendig.
      layout=durchmesser*1.0, #Der Multiplikator hinter der Variable, kann das Netzwerk
#vergrößern oder verkleinern Für einen Plot in R-Studio kann der Wert
#durchaus bei 3 liegen. Für die Ausgabe in der PDF sollte der Wert in der Regel
#nicht über 1,5 liegen.
      vertex.label.dist=0.08, #Abstand der Beschriftung zum jeweiligen Knoten
      vertex.frame.color=rgb(0,0,0,alpha=0.01), #Auch der Umrandung eines Knotens kann
#eine Farbe zugewiesen werden. Hier nutzen wir keinen festen Farbnamen wie "black"
#sondern die Farbgebung mit einem RGB-Wert
      #Die RGB -> Rot, Grün, Blau Werte werden hier in Prozent also 0 bis 1 angegeben
      #und nicht von 0-255 wie in der Bildverarbeitung üblich. Der Alpha-Kanal gibt an,
      #ob die Farbe durchsichtig sein soll. 0 bedeutet durchsichtig und 1 bedeutet
      #voll sichtbar.
      edge.width=0.15, #legt die Breite der Kanten fest. Für das Gesamtnetzwerk war
#der Wert von 2.5 viel zu dick.
      ) # Ende des Plot-Befehls für die erste grafische Darstellung des Gesamtnetzwerks
```

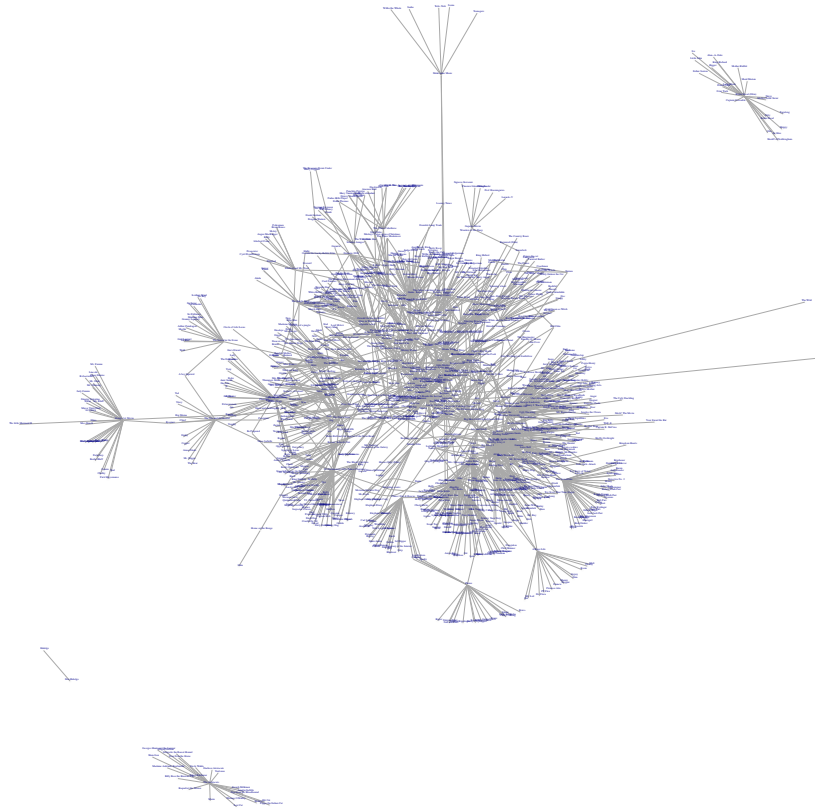


Abb. 2 - Gesamtnetzwerk mit einer ersten grafischen Anpassung.

Teilnetzwerke

Erstes Teilnetzwerk

Da sich in dem Gesamtnetzwerk keine Beziehungen erkennen lassen, muss das Netzwerk zunächst etwas eingedampft werden. Dazu entfernen wir sämtliche Knoten, die nur eine Verbindung aufweisen. Also Figuren, die in nur einem Film vorkommen oder Isolates sind. Dies erfolgt in zwei Schritten, da bei der Vereinfachung und Reduktion um Isolates und Einfachverbindungen neue Isolates entstehen können. Daher wird der selbe Befehl ein zweites Mal nachgeschaltet.

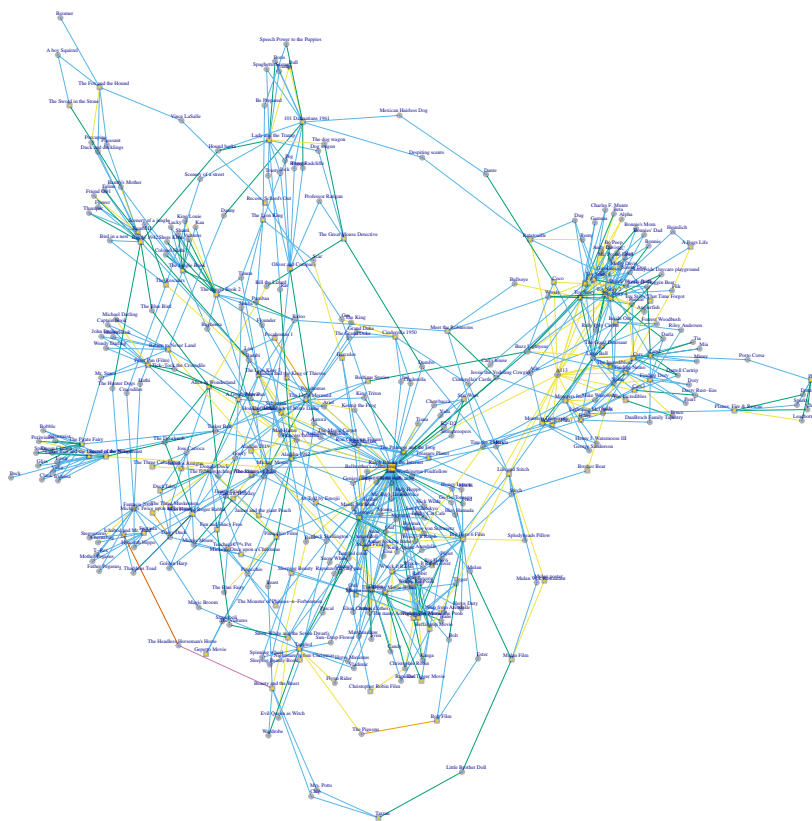


Abb. 3 - Disney Gesamtnetzwerk ohne Isolates und Einfachverbindungen

Dadurch lassen sich schon wesentlich mehr Knoten erkennen.

Um darzustellen, welche Figuren besonders wichtig sind, greifen wir bei der Darstellung der Cameo-Netze auf den Betweenness-Wert zurück. Dieser sagt aus, welche Figuren einen besonders hohen Stellenwert im Netzwerk einnehmen, da sie verschiedene Akteure und Teilnetze überhaupt erst verbinden. Je höher dieser Wert ist, umso mehr Akteure und Teilnetze verbindet er.

Bei der Analyse unserer Teilnetzwerke (Disney und Pixar) greifen wir auf diesen Wert zurück um die Größe der Knoten in Abhängigkeit ihrer Wichtigkeit zu variieren. Hierbei nutzen wir außerdem die Möglichkeiten der mathematischen Grundlagen der sogenannten affinen Transformation.

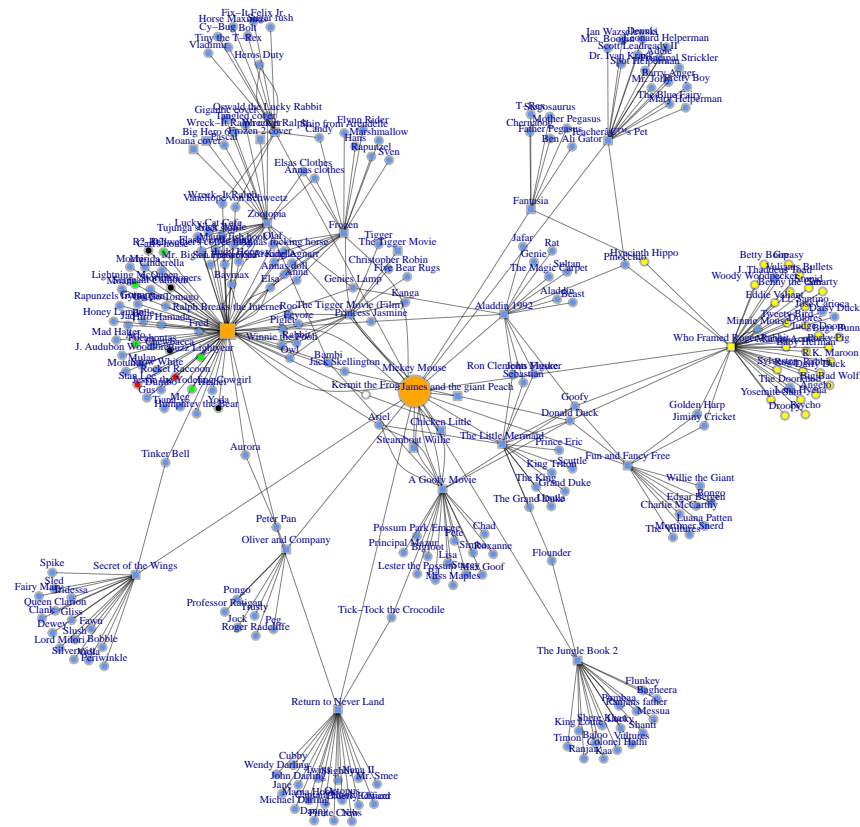
Hierbei lässt sich die maximale und minimale Größe der Knoten durch das Einsetzen verschiedener Werte und Multiplikatoren verändern. Eine Abhandlung von Prof. Dr. Markus Stroppel dazu, findet sich im Code Ordner unseres Github.

<https://github.com/tinomarst/pixar-226305/raw/master/code/affine-trafo.pdf>



Ego-Netzwerke spannender Akteure oder Filme

Im nächsten Schritt untersuchen wir die Ego-Netzwerke, die wir spannend fanden.

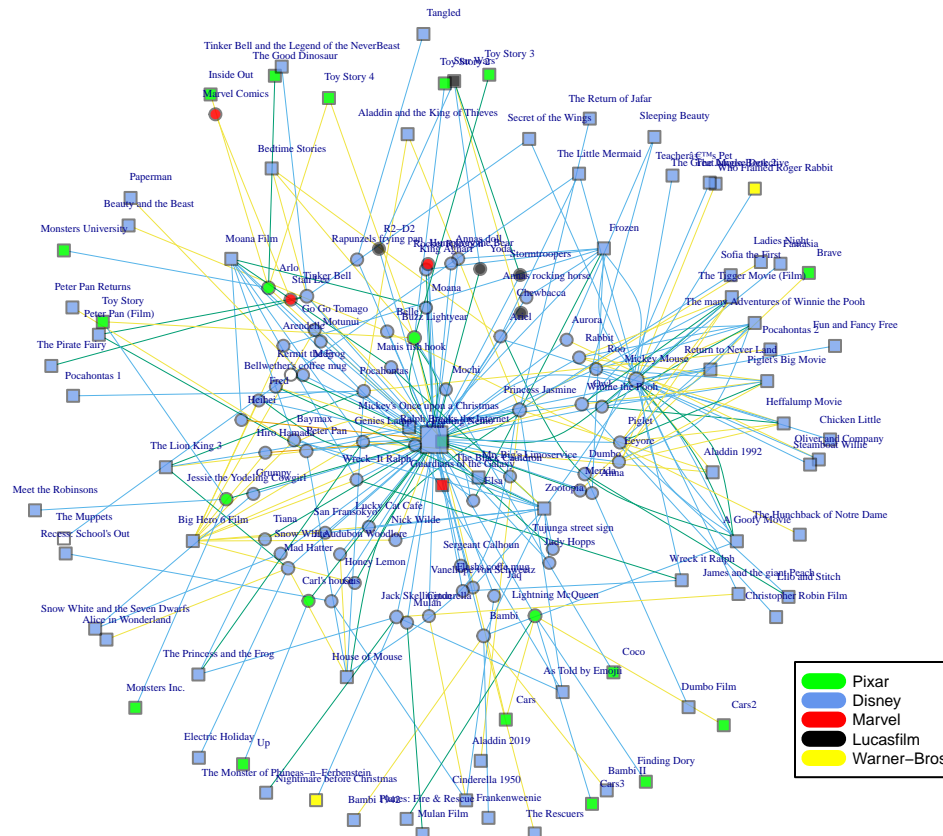


Ralph breaks the Internet

In der Voruntersuchung der Daten hat sich rbti schon als spannend herausgestellt







Beschränkung der Figuren nach den Produktionsstudios

Cameos nach Studios - wer produziert die meisten Easter Eggs

Pixar Teilnetzwerk

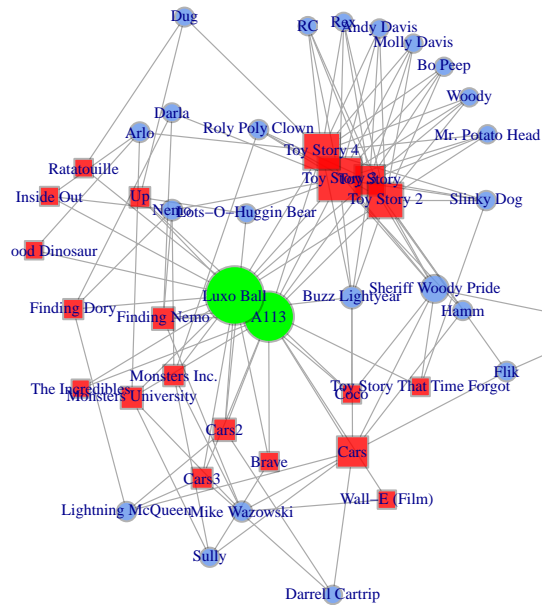
Disney Teilnetzwerk

Disney und Pixar Teilnetzwerk blabla

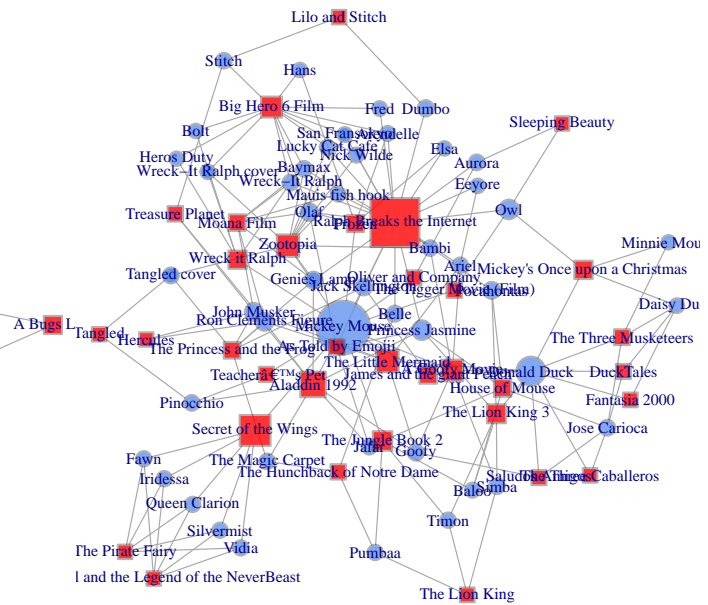
blubb

Plot der drei Netzwerke noch mal Blabla

Cameos bei Pixar

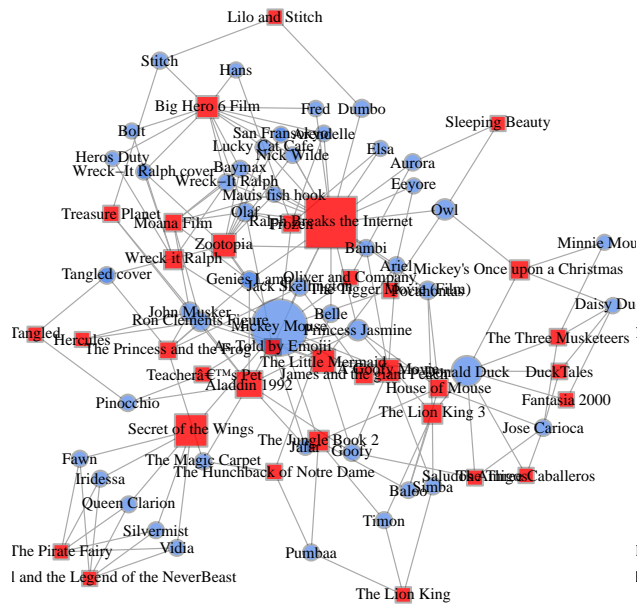


Cameos bei Disney

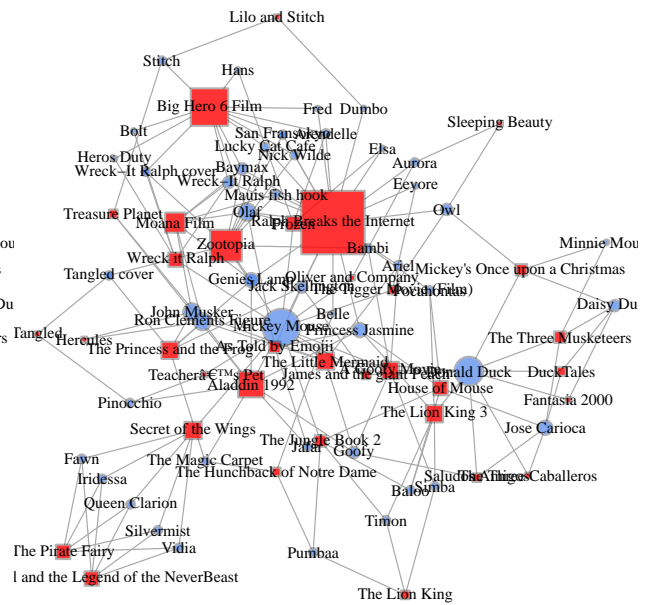


Wenn hier Text dazwischen steht?

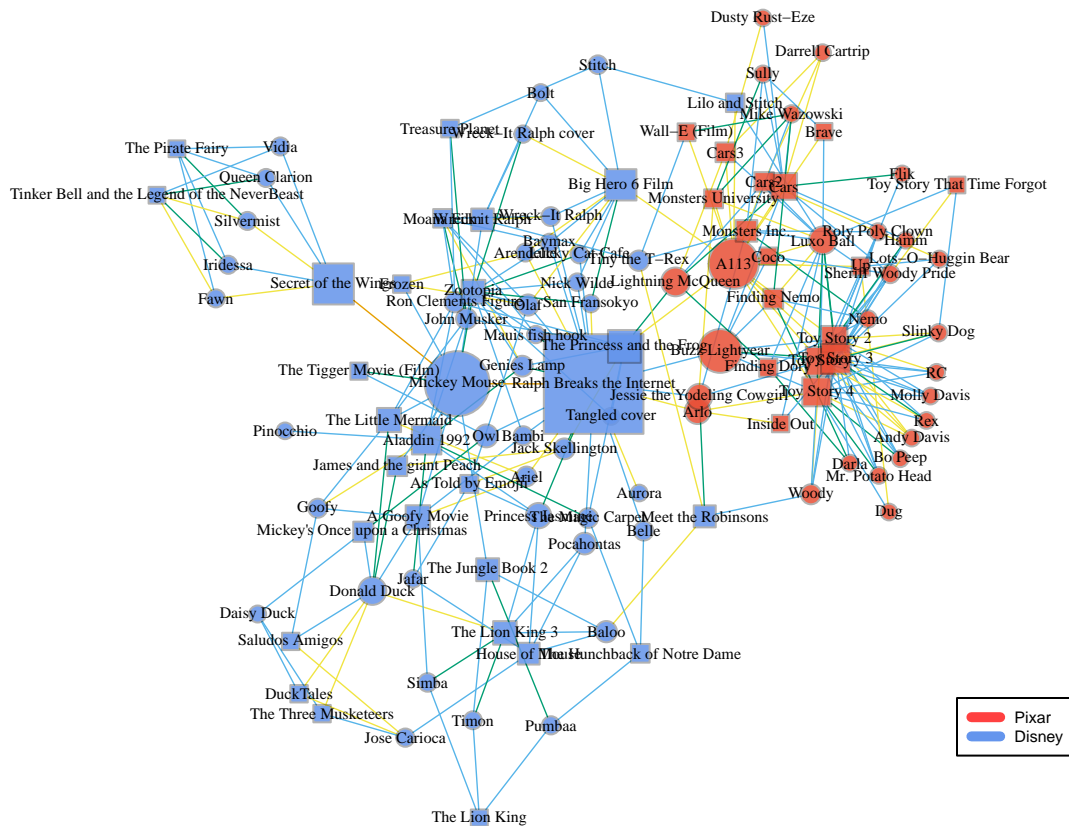
Disney Cameos Betweenness



Disney Cameos Degree

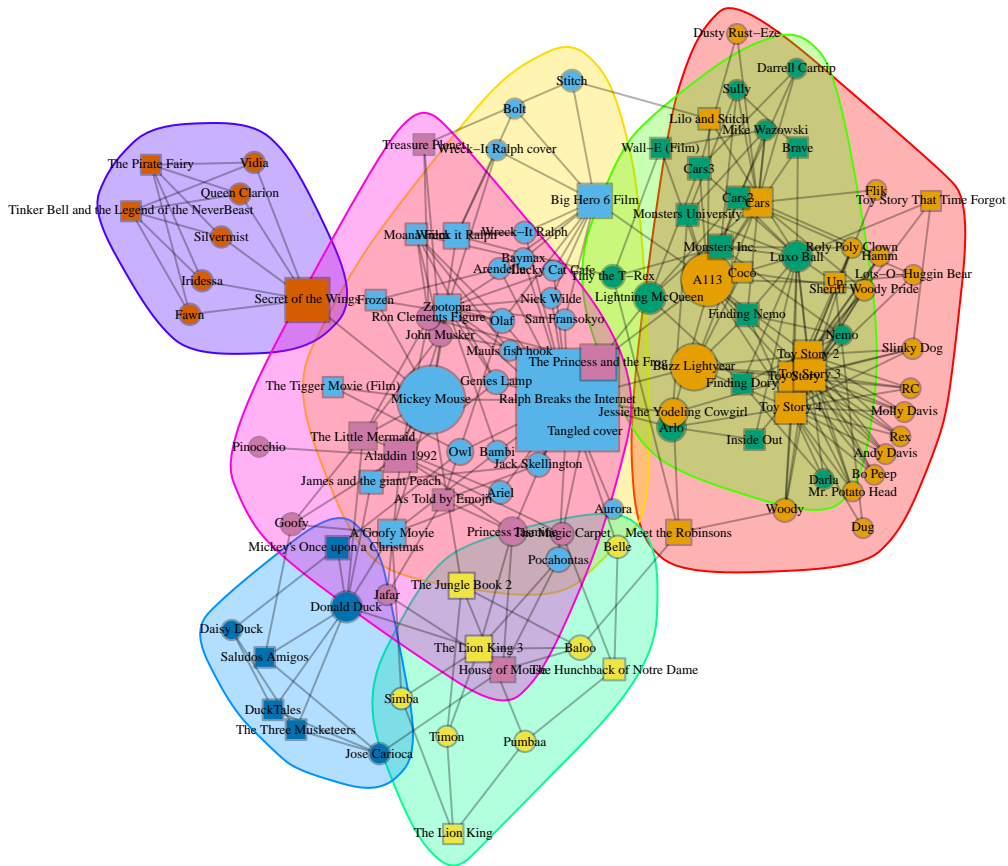


Hier könnte mehr Bla Bla stehen



```
closeness.cent <- closeness(dip2, mode="all")
closeness.cent
```

```
## [1] "communities"
```



Literaturverzeichnis

Fuhse, Jan Arendt (2018): *Soziale Netzwerke. Konzepte und Forschungsmethoden*. 2. Auflage. Konstanz, München: UVK Verlagsgesellschaft mbH.

Bondy, A., Murty, M. Ram. (2010). *Graph Theory*. London: Springer Verlag

Duden Redaktion. (o.J.). *Duden / Cameo / Rechtschreibung, Bedeutung, Definition, Herkunft*. Abgerufen 25. November 2019, von <https://www.duden.de/node/27392/revision/27421>

Filmstarts. (o. J.). *Das Geheimnis der „A113“: Die versteckten Easter Eggs in Pixar-Filmen!* Abgerufen 28. November 2019, von <http://www.filmstarts.de/filme/bildergalerien/bildergalerie-18500908/>

HarperCollins Publishers. (o.J.). *Easter egg Definition und Bedeutung / Collins Wörterbuch*. Abgerufen 28. November 2019, von <https://www.collinsdictionary.com/de/worterbuch/englisch/easter-egg>

Kaever, O. (2019, November 12). *Disney startet Streamingkanal: Kreative Komplettverweigerung*. Spiegel Online. Abgerufen von: <https://www.spiegel.de/kultur/tv/disney-streaming-kanal-bietet-wiederverwursthung-in-serie-a-1295956.html>

National Museum of American History (2018). *Mickey Mouse turns 90*. Abgerufen am: 25.11.2019 von: <https://americanhistory.si.edu/blog/mickey-mouse-turns-90>

o.V. (2014). *Geheimnis um Code in Disney-Filme gelüftet*. Abgerufen am: 25.11.2019 von: https://rp-online.de/kultur/film/a113-geheimnis-um-code-in-disney-filmen-gelueftet_aid-20408935

Petersen, R. (2012, Januar 18). *Cross-Over—Lexikon der Filmbegriffe*. Abgerufen 28. November 2019, von: <https://filmlexikon.uni-kiel.de/index.php?action=lexikon&tag=det&id=3092>

Anhang

Link zu unseren finalen Dateien auf Github

<https://github.com/tinomarst/pixar-226305>

Edgelist: https://raw.githubusercontent.com/tinomarst/pixar-226305/master/daten/el_fig.csv Nodelist: https://raw.githubusercontent.com/tinomarst/pixar-226305/master/daten/nl_fig.csv Code: <https://raw.githubusercontent.com/tinomarst/pixar-226305/master/code/Pixar-Universum-2.2a.Rmd> Codebuch: <https://github.com/tinomarst/pixar-226305/blob/master/Codebuch%202.0%20PB%202.10..pdf>

Poster: <https://raw.githubusercontent.com/tinomarst/pixar-226305/master/final/Plakat.jpg> Präsentation: <https://github.com/tinomarst/pixar-226305/raw/master/final/Pr%C3%A4sentation%20Netzwerkanalyse%20112619.pdf>

Aufstellung der Aufgaben pro Team-Mitglied

Nachfolgend werden zur Übersichtlichkeit die Rufnamen der Teammitglieder verwendet.

Milena: Milena Sager, Pauline: Pauline Bort, Tino: Justin Marquez, Felix: Felix Arnold, Conny: Constanze Kuba

Aufgabe	Teammitglied	Zeitaufwand p.P.
Idee	Milena	2h
Forschungsfrage und Design	Team	4h
Präsentation im 3. Semester	Felix, Tino	1h
Datenerhebung	Team	20h
ID-Vergabe	Tino, Felix, Milena, Pauline	2h
Erstellung Edge- und Nodelist	Team	5 h
Datenkorrektur/ -bereinigung: Datensatz zu Beginn, sowie Edge- und Nodelist	Tino,Felix, Milena, Pauline	10h
erste Netzwerkerstellung (Blocktermin)	Tino, Pauline, Felix, Milena	10h
Netzwerkerstellung/Skript	Tino	54+ h
Netzwerkmaße, Überarbeitung Skript	Tino, Milena	6h
weitere Edge- und Nodelist für Pixar mit Recherche	Milena	6h
Konzept Abschlusspräsentation	Conny, Milena, Pauline	2h
Finale Datenkorrekturen	Felix, Tino	2h
Abschlusspräsentation: Erstellung und Präsentation	Conny, Milena	10h
Plakat	Pauline	?????
Forschungsbericht	Milena	8h
Forschungsbericht	Pauline	6h
Forschungsbericht	Tino	5h
Forschungsbericht	Conny	10h
Forschungsbericht	Felix	4h

Workload gesamt	Stunden
Conny	
Felix	
Tino	viel
Milena	85 h
Most-valuable Team-Member	Tino da höchster Zeitaufwand und sehr engagiert und geduldig mit uns und R